

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
12 April 2001 (12.04.2001)

PCT

(10) International Publication Number
WO 01/25991 A1

(51) International Patent Classification⁷: G06F 17/60

Lane, Princeton Junction, NJ 08550 (US). SHALL-CROSS, David; Apt. 1, 44 Center Grove Road, Randolph, NJ 07869 (US).

(21) International Application Number: PCT/US00/26301

(22) International Filing Date:
25 September 2000 (25.09.2000)

(74) Agents: GIORDANO, Joseph et al.; International Coordinator, Telcordia Technologies, Inc., 445 South Street, Rm. 1G112R, Morristown, NJ 07960-6438 (US).

(25) Filing Language: English

(81) Designated States (*national*): CA, JP.

(26) Publication Language: English

(84) Designated States (*regional*): European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

(30) Priority Data:
09/411,014 1 October 1999 (01.10.1999) US

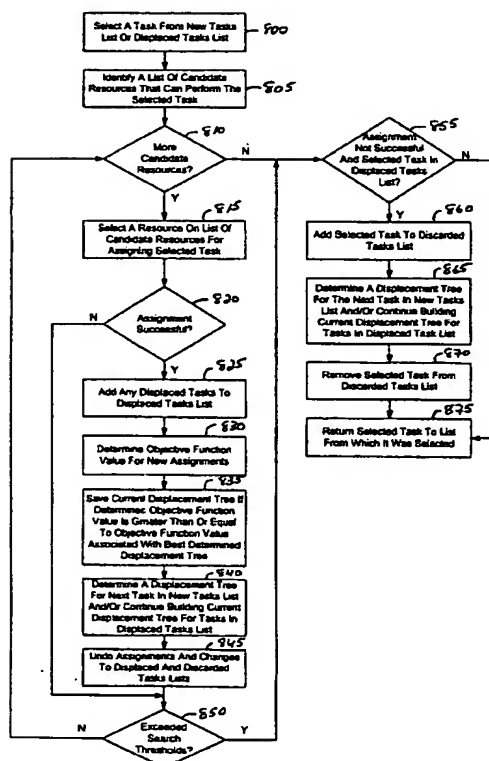
(71) Applicant: TELCORDIA TECHNOLOGIES, INC.
[US/US]; 445 South Street, Morristown, NJ 07960-6438 (US).

Published:
— With international search report.

(72) Inventors: CASEAU, Yves; 31, avenue Mirabeau, F-78000 Versailles (FR). KOPPSTEIN, Peter; 7 Suffolk

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND SYSTEM FOR ASSIGNMENT OF TASKS TO RESOURCES USING DISPLACEMENT TREES



(57) Abstract: A method and system assign new tasks to a set of resources that have a first plurality of assignments by determining one or more displacement trees. A scheduling system determines, subject to one or more predetermined thresholds (850), a displacement tree (800) and displaces from the first plurality of assignments one or more of the set of assigned tasks according to the displacement tree. A displacement tree (800) may include zero or any number of assigned tasks that the scheduling system must displace from the first plurality of assignments, for example unassign and/or reassign, to assign the new tasks to the set of resources. The scheduling system then assigns the new tasks to the set of resources (805, 815), such that an objective function value (830) of second plurality of assignments that include the assigned new tasks is greater than or equal to an objective function value of the first plurality of assignments.

Best Available Copy

WO 01/25991 A1

IAP5 Rec'd PCT/PTO 29 MAR 2006

**METHOD AND SYSTEM FOR ASSIGNMENT OF TASKS
TO RESOURCES USING DISPLACEMENT TREES**

5 CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to U.S. Application No. 09/126168, filed July 30, 1998, the contents of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

10 The present invention relates generally to vehicle routing and scheduling problems, and more particularly, to a method and system for assigning tasks to resources using displacement trees.

A fundamental problem in any vehicle routing and scheduling problem is to assign a set of tasks to a set of resources so as to optimize an objective function value associated with the assignments. An objective function may be determined based on, for example, the number of tasks assigned to the set of resources, the total weight of the assigned tasks as determined by, for example, the priorities of the assigned tasks, the total duration of the assigned tasks, the number of assigned tasks that are assigned to preferred resources as determined by, for example, the qualifications of the resources, the number of deadlines missed because of unassigned tasks, and/or the total travel time for dispatching a resource to perform an assigned task.

20 Generally, in any vehicle routing and scheduling problem, each resource has an assignment list (or a schedule), which identifies the tasks assigned to the resource and the scheduled start times and completion times for the assigned tasks. When new tasks must be performed, the new tasks are assigned to resources whose schedules have sufficient idle time between assigned tasks.

A simple method would be to insert the new tasks into schedules that have idle times between completion time of an assigned task and the start time of the next assigned task. The problem with this method, however, is that the duration of a new task may not allow the new task to be assigned to a resource, even though the resources collectively have sufficient idle time in their schedules. For example, consider two resources A and B each having a schedule that includes a one hour task

30

and an hour of idle time. Using this method, a new task that has a duration of two hours cannot be assigned to either resources A or B, even though resources A and B collectively have two hours of idle time.

More sophisticated methods, however, displace a single task from each resource's schedule in a sequential fashion until the new task can be assigned to a resource. Applying such methods to the above example, a new task that has a duration of two hours can be assigned to either resource A or B by displacing one of the one hour tasks from one resource to another. For example, the one hour task from resource A may be displaced to resource B's schedule, allowing the new task to be assigned to resource A. The problem with these methods, however, is that the objective function value associated with the new assignments may sometimes be less than the objective function value associated with the prior assignments.

Therefore, it is desirable to have a method and system for assigning tasks to resources using displacements that address the above and other disadvantages of the prior art.

DESCRIPTION OF THE INVENTION

Methods and systems consistent with the present invention assign a new task to a set of resources that have a first plurality of assignments by determining, subject to one or more predetermined thresholds, a displacement tree. A new task may be, for example, any task that is not assigned to any of the set of resources. A displacement tree may include zero or any number of assigned tasks that must be displaced from the first plurality of assignments to assign the new task to a resource in the set of resources.

Such methods and systems then displace from the first plurality of assignments one or more assigned tasks according to the displacement tree, and assign the new task to a resource, such that an objective function value associated with a second plurality of assignments that include the assigned new task is greater than or equal to an objective function value associated with the first plurality of assignments.

An objective function may be determined based on, for example, the number of tasks assigned to the set of resources, the total weight of the assigned tasks as determined by, for example, the priorities of the assigned tasks, the total duration of

the assigned tasks, the number of assigned tasks that are assigned to preferred resources as determined by, for example, the qualifications of the resources, the number of deadlines missed because of unassigned tasks, and/or the total travel time for dispatching a resource to perform an assigned task.

- 5 The description of the invention and the following description for carrying out the best mode of the invention should not restrict the scope of the claimed invention. Both provide examples and explanations to enable others to practice the invention. The accompanying drawings, which form part of the description for carrying out the best mode of the invention, show several embodiments of the invention, and together
10 with the description, explain the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the Figures:

- Figure 1 is an external block diagram of a scheduling system for
15 assigning M tasks to N resources, in accordance with an embodiment of the present invention;

Figure 2 is an internal block diagram of a scheduling system for assigning M tasks to N resources, in accordance with an embodiment of the present invention;

- 20 Figure 3 is a block diagram of a scheduling module in a scheduling system, in accordance with an embodiment of the present invention;

Figure 4 illustrates a resources table in a scheduling module, in accordance with an embodiment of the present invention;

- 25 Figure 5 illustrates assignments lists associated with resources, in accordance with an embodiment of the present invention;

Figure 6 is a flow chart of the steps performed by a scheduling program for assigning an unassigned task to a set of resources, in accordance with an embodiment of the present invention;

- 30 Figures 7a, 7b, 7c, and 7d illustrate a set of resources and associated assignments before and after assignment of a new task to the set of resources based on a displacement tree, in accordance with an embodiment of the present invention; and

Figure 8 is a flow chart of the steps performed by a scheduling

program for determining a displacement tree, in accordance with an embodiment of the present invention.

BEST MODE FOR CARRYING OUT THE INVENTION

5 Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

 In accordance with an embodiment of the present invention, a scheduling
10 system assigns new tasks to a set of resources that have a first plurality of assignments by determining one or more displacement trees. A resource may be, for example, a technician, an equipment, and/or any entity that can perform a task. Each assignment includes a set of assigned tasks, which are defined by, for example, a task identifier, a start time, a completion time, and a description of what has to be performed.

15 When a new task must be performed, the scheduling system determines, subject to one or more predetermined thresholds, a displacement tree and displaces from the first plurality of assignments one or more of the set of assigned tasks according to the displacement tree. A displacement tree may include zero or any number of assigned tasks that the scheduling system must displace from the first
20 plurality of assignments, for example unassign and/or reassign, to assign the new task to the set of resources.

 The scheduling system then assigns the new tasks to the set of resources, such that an objective function value associated with a second plurality of assignments that include the assigned new tasks is greater than or equal to an objective function value
25 associated with the first plurality of assignments. An objective function may be determined based on, for example, the number of tasks assigned to the set of resources, the total weight of the assigned tasks as determined by, for example, the priorities of the assigned tasks, the total duration of the assigned tasks, the number of assigned tasks that are assigned to preferred resources as determined by, for example,
30 the qualifications of the resources, the number of deadlines missed because of unassigned tasks, and/or the total travel time for dispatching a resource to perform an assigned task.

Figure 1 is an external block diagram of a scheduling system 100 for assigning M tasks to N resources, where M and N are integers greater than zero, in accordance with an embodiment of the present invention. As shown, scheduling system 100 interfaces with desktop computers 101 and 102, monitors 110 and 111, and printer 112 via connections 103, 104, 113, 114, and 115, respectively. Via desktop computers 101 and 102, a user may enter and modify in scheduling system 100 tasks, resources, and assignments associated with the tasks and the resources. Furthermore, scheduling system 100 dynamically assigns tasks to resources and vice versa, and, if necessary, modifies the assignments that are entered by users.

In the embodiment of Figure 1, scheduling system 100 includes N resources, M tasks, and associated assignments, all of which are governed by various timing rules, for example, specific time windows for completing a task. Scheduling system 100 may assign a task to zero or one resource. A task that is not assigned to any resource is referred to as an "unassigned" or "pending" task.

Figure 2 is an internal block diagram of scheduling system 100, in accordance with an embodiment of the present invention. As shown, scheduling system 100 comprises a processor 200, which connects via a high speed bus 250 to a memory 210, an input device 220, an output device 230, and a secondary storage 240.

Memory 210 includes scheduling module 212 and operating system 214, both of which may include data and a set of instructions in the form of software that processor 200 executes.

Input device 220 may include, for example, a standard terminal or a desktop computer, from which users may input information associated with resources, tasks, and/or assignments. Output device 230 may include, for example, a terminal and a printer for displaying task, resource, and/or assignment information.

Secondary storage 240 comprises a disk drive and a tape drive or CD Read Only Memory (ROM). From the tape drive or CD ROM, software and data may be loaded onto the disk drive, which can then be copied into memory 210. Similarly, software and data in memory 210 may be copied onto the disk drive, which can then be loaded onto the tape drive or CD ROM.

Figure 3 is a block diagram of scheduling module 212 in memory 210, in accordance with an embodiment of the present invention. As shown, scheduling

module 212 includes a scheduling program 300, a Resources Table 310, an Assignments Lists 320, a New Tasks List 330, a Displaced Tasks List 340, and a Discarded Tasks List 350.

Figure 4 illustrates Resources Table 310, in accordance with an embodiment of the present invention. Resources Table 310 includes N entries 400₁-400_N, where each entry is associated with a resource. Each entry 400₁-400_N includes a resource identifier and resource attributes. A resource identifier uniquely identifies a resource, and may include, for example, an alphanumeric string. A resource attribute may include, for example, a list of attributes that are associated with the resource. The attributes may include, for example, the availability and qualifications of the resource.

Figure 5 is a block diagram of Assignments Lists 320, in accordance with an embodiment of the present invention: Assignments Lists 320 includes an array of N lists 500₁-500_N, which are associated with N resources, respectively. As shown, a list 500₁-500_N may include, for example, a plurality of nodes, each of which includes information about a task that is assigned to a resource. A node may include, for example, a task identifier, which uniquely identifies a task, a priority, a scheduled start time, and a scheduled end time. When scheduling program inserts a task into a list in Assignments Lists 320, the task is referred to as an "assigned" task.

New Tasks List 330 may include, for example, a list of tasks that are not assigned to any resources. Scheduling program 300 adds to New Tasks List 330 new unassigned tasks, which are, for example, entered by users via input device 220. New Tasks List 330 may, for example, be organized according to the priorities of the new tasks.

Displaced Tasks List 340 may include, for example, a list of tasks that scheduling program 300 displaces, for example removes, from Assignments Lists 320. When scheduling program 300 displaces a task from a list in Assignments Lists 320, the task is referred to as an "unassigned" task.

Discarded Tasks List 350 may include, for example, a list of tasks that scheduling program 300 displaces from a list in Assignments Lists 320, but cannot subsequently reassign to any of the N resources. When scheduling program 300 inserts a task into Discarded Tasks List 350, the task is referred to as a "discarded" task.

Figure 6 is a flow chart of the steps performed by scheduling program 300 for assigning a new unassigned task to a set of resources, in accordance with an embodiment of the present invention. Scheduling program 300 determines whether there are any tasks in New Tasks List 330 (step 600). When scheduling program 300
5 determines that there is at least one task in New Tasks List 330, scheduling program 300 initializes Displaced Tasks List 340 and Discarded Tasks List 350 by, for example, setting each list to a null pointer (step 610).

As shown below in detail, scheduling program 300 then determines a displacement tree for assigning a new task in New Tasks List 330 to a resource (step
10 620). A displacement tree identifies zero or more assigned tasks that scheduling program 300 must, for example, unassign and/or reassign in order to assign the selected new task to a resource.

For example, Figures 7a-d illustrate the assignment of a new task E to a set of resources A, B, C, and D based on a displacement tree. Figure 7a shows resources A,
15 B, C, and D and their respective assigned tasks before scheduling program 300 assigns task E, which resides in New Tasks List 330, to one of the resources A-D. As shown, tasks K, F, and G are assigned to resource A; tasks H and I are assigned to resource B; task J is assigned to resource C, and task L is assigned to resource D.

As shown by dotted arrows, to assign task E to resource A, scheduling
20 program 300 displaces tasks F and G from resource A's assignments. To reassign the displaced task F to resource B, scheduling program 300 then displaces tasks H and I from resource B's assignments. To reassign the displaced task G to resource C, scheduling program 300 displaces task J from resource C's assignments. Finally, to reassign the displaced tasks H and I to resources D and A, respectively, scheduling
25 program 300 displaces tasks L and K from resource D's and A's assignments, respectively.

The above steps of displacing the assigned tasks F, G, H, I, J, L, and K may be represented as a displacement tree, which is shown in Figure 7b. From the set of displaced tasks F, G, H, I, J, L, and K, scheduling program 300 reassigns tasks F, G,
30 H, and I to resources B, C, D, and A, respectively. As shown in Figure 7c, scheduling program 300, however, adds the displaced tasks J, K, and L to Displaced Tasks List 340, as these tasks are not reassigned.

Figure 7d shows the new assignments resulting from displacements of tasks F, G, H, I, J, L, and K. As shown, tasks I and E are assigned to resource A; task F is assigned to resource B, task G is assigned to resource G; and task H is assigned to resource D.

- 5 Referring back to Figure 6, after determining a displacement tree (step 610), scheduling program 300 determines an objective function value associated with the new assignments resulting from the determined displacement tree. Scheduling program 300 then determines whether the determined objective function value associated with the new assignments is greater than or equal to the objective function value associated with the current assignments (step 620).

The objective function value may be determined based on, for example, the number of tasks assigned to the set of resources, the total weight of the assigned tasks as determined by, for example, the priorities of the assigned tasks, the total duration of the assigned tasks, the number of assigned tasks that are assigned to preferred resources as determined by, for example, the qualifications of the resources, the number of deadlines missed because of unassigned tasks, and/or the total travel time for dispatching a resource to perform an assigned task.

If scheduling program 300 determines that the determined objective function value associated with the new assignments is not greater than or equal to the objective function value associated with the current assignments, scheduling program 300 removes the selected new task from New Tasks List 330 (step 640).

If scheduling program 300 determines that the determined objective function value associated with the new assignments is greater than or equal to the objective function value associated with the current assignments, scheduling program 300 sets the current assignments to the new assignments by modifying Assignments Lists 320 according to the new assignments (step 650).

Scheduling program 300 then determines the number of tasks in New Tasks List 330 that are assigned to resources based on the determined displacement tree (step 660). Finally, scheduling program 300 transfers one or more tasks from Displaced Tasks List 340 and/or Discarded Tasks List 350 to New Tasks List 330, where the number of transferred tasks equals, for example, one less than the determined number of tasks assigned in step 660 (step 670).

Finally, scheduling program 300 repeats steps 600-670 for the next task in New Tasks Lists 330 until there are no more tasks in New Tasks Lists (step 605).

Figure 8 is a flow chart of the steps performed by scheduling program 300 for determining a displacement tree (shown as step 610 in Figure 6), in accordance with an embodiment of the present invention. Scheduling program 300 selects a task from New Tasks List 330 or Displaced Tasks List 800 (step 800). Scheduling program 300 then identifies a list of candidate resources that can perform the selected task (step 805). The candidate resources may be identified based on, for example, the qualifications of the resources and/or some function of the relation of the selected task, for example geographical proximity, to other tasks assigned to the candidate resources.

For each identified candidate resource (step 810), scheduling program 300 performs steps 815-850 as described below. Scheduling program 300 selects a candidate resource from the candidate resource list (step 815). Scheduling program 300 then determines whether the selected task can be assigned to the selected candidate resource by possibly displacing one or more assigned tasks from the selected candidate resource's assignments (step 820).

If scheduling program 300 determines that the selected task can be assigned to the selected candidate resource, scheduling program 300 adds any resulting displaced tasks from the candidate resource's assignments to Displaced Tasks Lists 340 (step 825). Otherwise, if scheduling program 300 determines that the selected task cannot be assigned to the selected candidate resource, scheduling program 300 determines whether one or more predetermined search thresholds are exceeded (step 850).

The search thresholds may include, for example, a limit on the number of candidate resources scheduling program 300 considers before assigning to a resource a selected task from New Tasks List 330, a limit on the number of tasks in Displaced Tasks List 340, a limit on the depth of each displacement tree, a limit on the number of tasks displaced from Assignments Lists 320 by a selected task, and/or a limit on the total duration of assigned tasks displaced from Assignments Lists 320 by a selected task.

After adding any resulting displaced tasks from the candidate resource's assignments to Displaced Tasks Lists 340 (step 825), scheduling program 300

determines an objective function value associated with the new assignments resulting from assigning the selected task to the selected candidate resource (step 830). If the determined objective function value is greater than or equal to the objective function value associated with the best displacement tree determined up to this step, scheduling
5 program 300 saves the current displacement tree, which includes one or more assigned tasks that are to be displaced from the selected candidate resources's assignments to Displaced Tasks List 340 (step 835).

Scheduling program 300 then determines a displacement tree for the next task in New Tasks List 330 and/or continues to build the current displacement tree for the
10 next task in Displaced Tasks List 340, subject to the above-mentioned predetermined search thresholds (step 840). Scheduling program 300 may determine the displacement tree and continue building the current displacement tree by, for example, recursively performing the steps 800-875. After performing one or more recursions, scheduling program 300 has determined a displacement tree for assigning the selected
15 task to the selected candidate resource and for displacing zero or more assigned tasks from Assignment Lists 320.

After recursively performing the steps 800-875, scheduling program 300 backtracks any new assignments made up to this step based on the current displacement tree (step 845). For example, scheduling program 300 unassigns any
20 new assignments made based on the current displacement tree, and restores Displaced Tasks List 340 and Discarded Tasks List 350 to their respective state prior to recursively performing the steps 800-875.

Scheduling program 300 then determines whether it has exceeded one or more of the predetermined search thresholds (step 850). For example, if scheduling
25 program 300 determines the number of candidate resources that are considered for the selected task does not exceed a predetermined threshold, scheduling program 300 considers the next candidate resource for assigning the selected task, and performs the steps 815-850 as described above.

If scheduling program 300 determines number of candidate resources that are
30 selected for the selected task exceeds the predetermined threshold and the selected task has not been successfully assigned to a candidate resource, scheduling program 300 determines whether the selected task is in Displaced Tasks List 350 (step 855). If

scheduling program 300 determines that the selected task is not in Displaced Tasks List 350, scheduling program 300 returns the selected task to the list from which it was selected (step 875).

5 If scheduling program 300 determines that the selected task is in Displaced Tasks List 350, scheduling program 300 adds the selected task to Discarded Tasks List 350 (step 860). Scheduling program 300 then determines a displacement tree for the next task in New Tasks List 330 and/or continues to build the current displacement tree for the next task in Displaced Tasks List 340 by, for example recursively performing the steps 800-875, subject to the predetermined search
10 thresholds (step 865).

After recursively performing the steps 800-875, scheduling program 300 removes the selected task from Discarded Tasks List 350 (step 870). Finally, scheduling program 300 returns the selected task to the list from which it was selected at step 800.

15 While it has been illustrated and described what are at present considered to be preferred embodiments and methods of the present invention, it will be understood by those skilled in the art that various changes and modifications may be made, and equivalents may be substituted for elements thereof without departing from the true scope of the invention.

20 In addition, many modifications may be made to adapt a particular element, technique or implementation to the teachings of the present invention without departing from the central scope of the invention. Therefore, it is intended that this invention not be limited to the particular embodiments and methods disclosed herein, but that the invention include all embodiments falling within the scope of the
25 appended claims.

CLAIMS

1. A method for assigning a task to a set of resources having a first plurality of assignments that include a set of assigned tasks, said method comprising the steps of:
 - 5 determining, subject to one or more predetermined thresholds, a displacement tree;
displacing from the first plurality of assignments one or more of the set of assigned tasks according to the displacement tree; and
assigning the task to one of the set of resources, such that an objective
 - 10 function value associated with a second plurality of assignments that include the assigned task is greater than or equal to an objective function value associated with the first plurality of assignments.
2. The method of claim 1, wherein the determining step comprises the
 - 15 steps of:
selecting one or more of the set of resources;
determining a set of displacement trees based on the selected resources, respectively;
determining new sets of assignments based on the set of displacement
 - 20 trees, respectively;
determining objective function values associated with the sets of new assignments, respectively; and
selecting from the set of displacement trees a displacement tree with the highest objective function value.
3. The method of claim 1, wherein the determining step comprises the
 - 25 steps of:
a) selecting a first one of the set of resources;
b) assigning the task to the selected resource, such that one or more of
 - 30 the assigned tasks that are associated with the selected first resource are displaced;
c) building a first displacement tree that includes the displaced tasks;
d) determining an objective function value associated with the first

displacement tree;

e) storing the first displacement tree when the determined objective function value is greater than or equal to an objective function value associated with a second displacement tree;

5 f) recursively assigning the displaced tasks to the set of resources and expanding the first displacement tree based on the assigning of the displaced tasks; and

g) backtracking assignments that are associated with the recursively built first displacement tree.

10

4. The method of claim 1, wherein the displacing step comprises the steps of:

unassigning one or more of the set of assigned tasks; and

reassigning one or more of the unassigned tasks to the set of resources,

15 such that the objective function value associated with the reassigned tasks and the assigned task is greater than or equal to the objective function value associated with the set of assigned tasks.

5. The method of claim 3 further comprising the steps of:

20 after the backtracking step, selecting a second one of the set of resources; and

performing the steps b, c, d, e, f, and g based on the selected second resource.

25 6. A system for assigning a task to a set of resources having a first plurality of assignments that include a set of assigned tasks, comprising:

a scheduling program for determining, subject to one or more predetermined thresholds, a displacement tree, and for displacing from the first plurality of assignments one or more of the set of assigned tasks according to the displacement tree, and for assigning the task to one of the set of resources, such that
30 an objective function value associated with a second plurality of assignments that include the assigned task is greater than or equal to an objective function value

associated with the first plurality of assignments; and
a processor for executing the scheduling program.

7. A computer-readable medium capable of configuring a computer to
- 5 perform a method for assigning a task to a set of resources having a first plurality of assignments that include a set of assigned tasks, said method comprising the steps of:
- determining, subject to one or more predetermined thresholds, a displacement tree;
- displacing from the first plurality of assignments one or more of the set
- 10 of assigned tasks according to the displacement tree; and
- assigning the task to one of the set of resources, such that an objective function value associated with a second plurality of assignments that include the assigned task is greater than or equal to an objective function value associated with the first plurality of assignments.

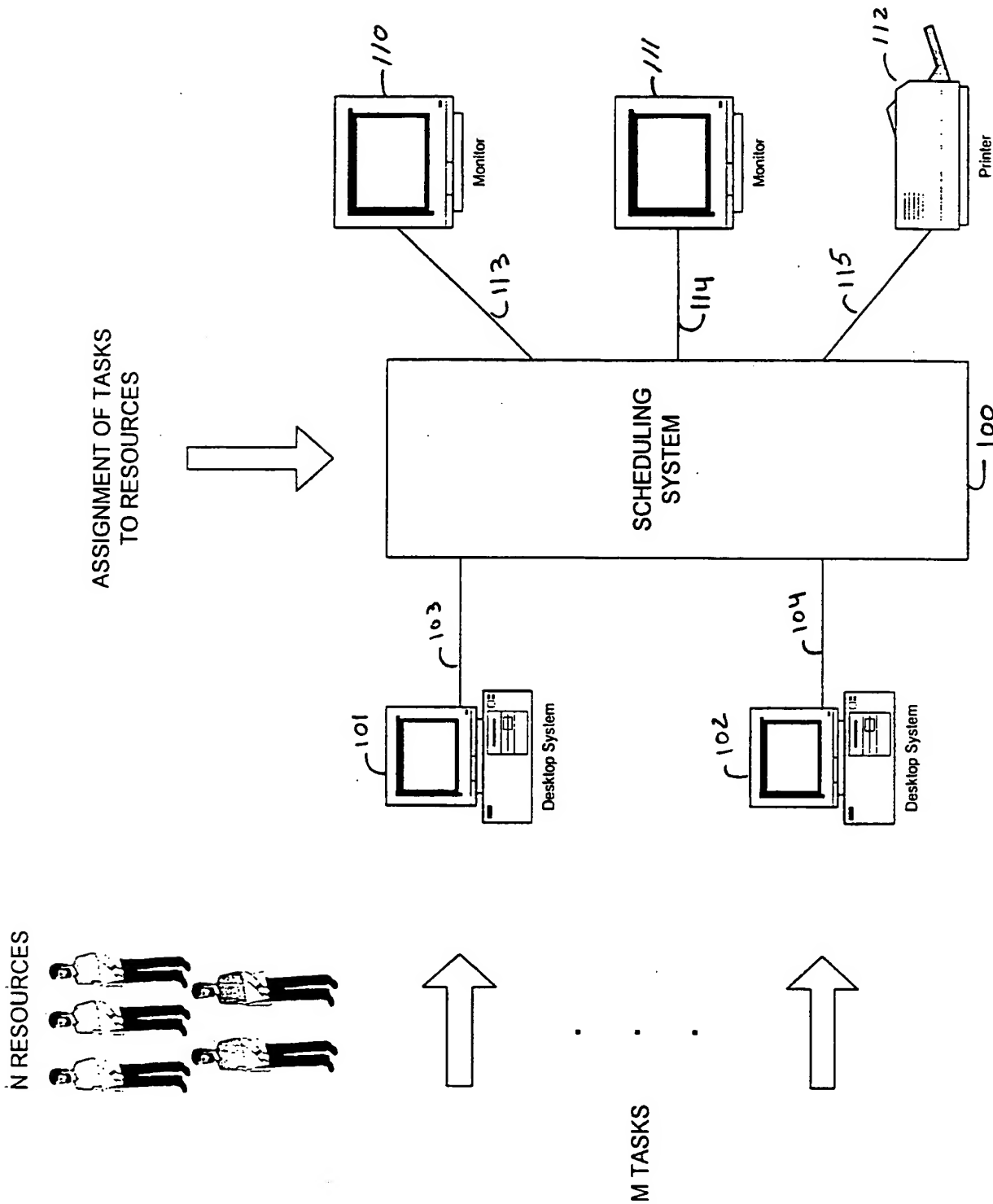


FIGURE 1

100 ~

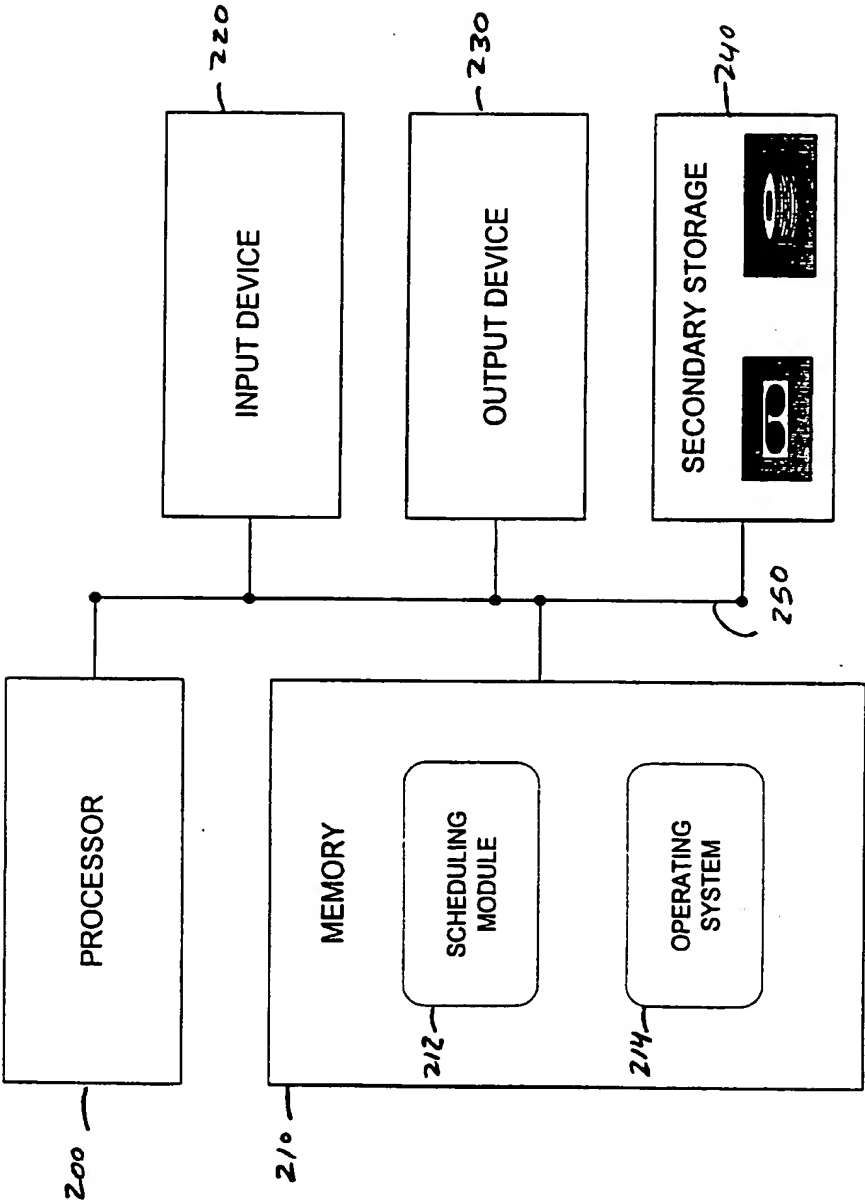


FIGURE 2

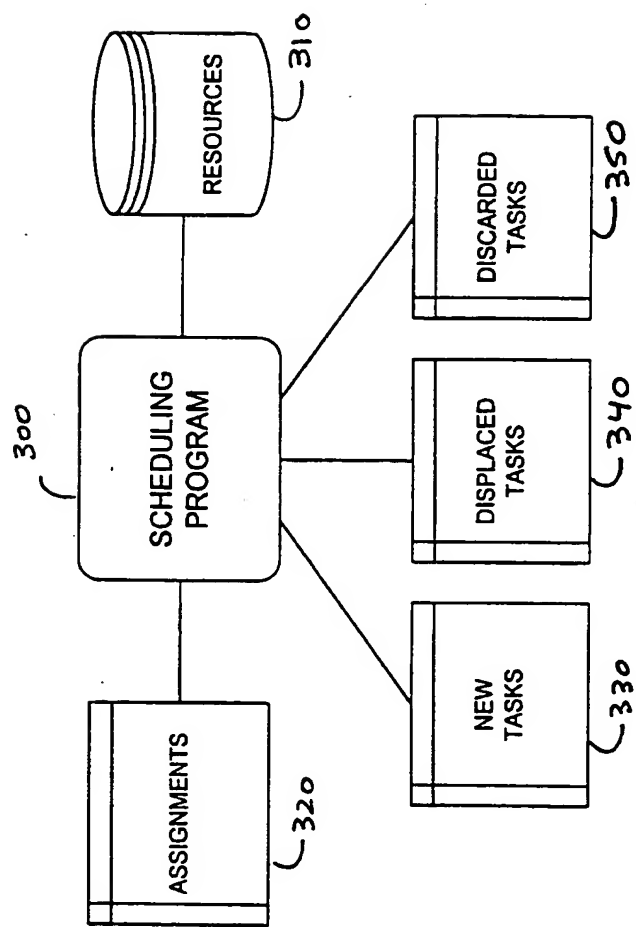


FIGURE 3

310 ~

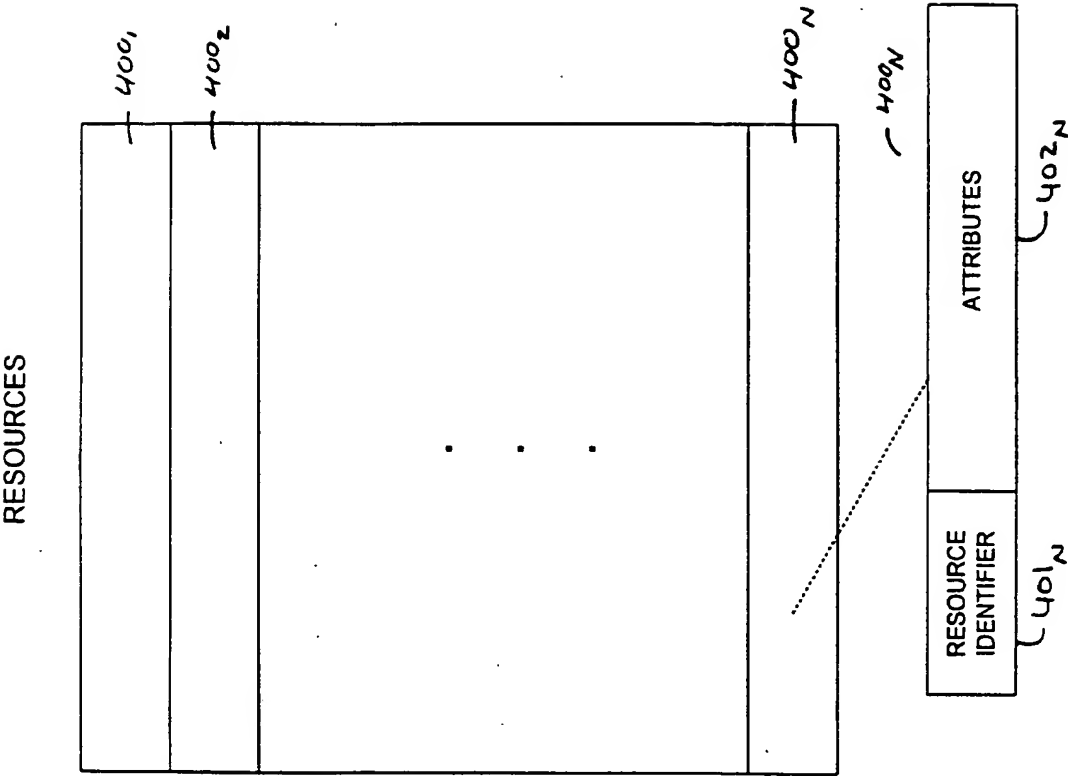


FIGURE 4

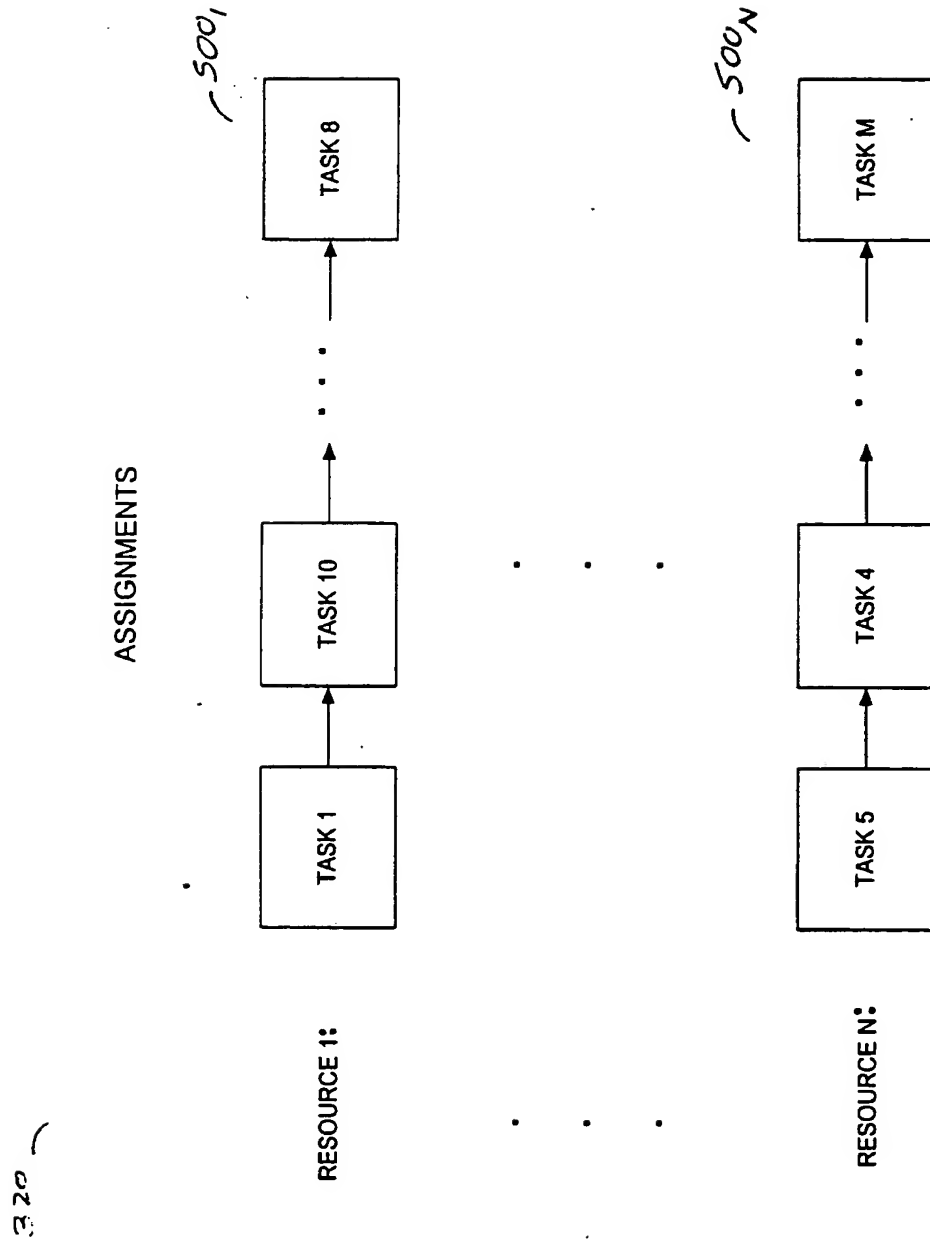


FIGURE 5

6/8

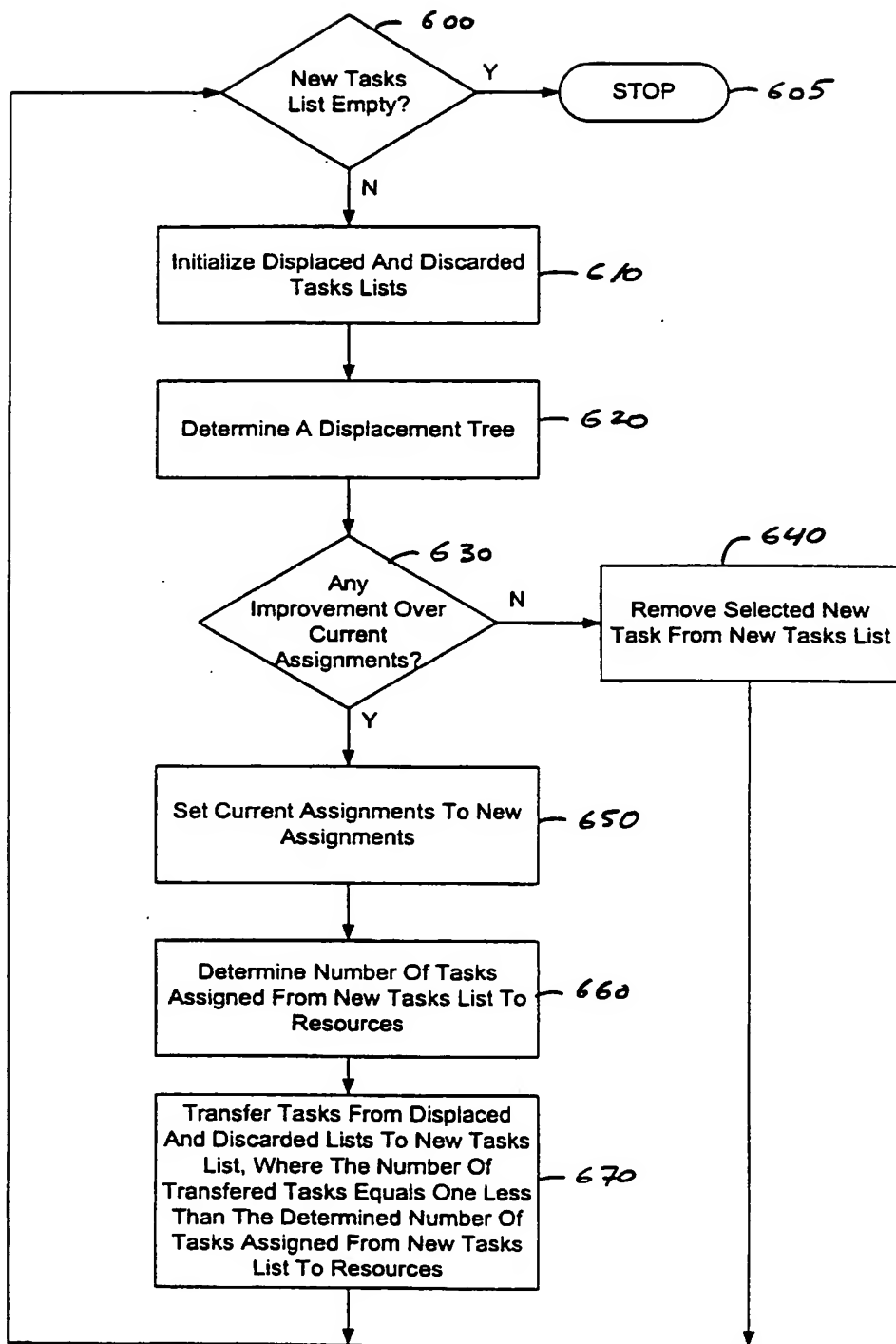


FIGURE 6

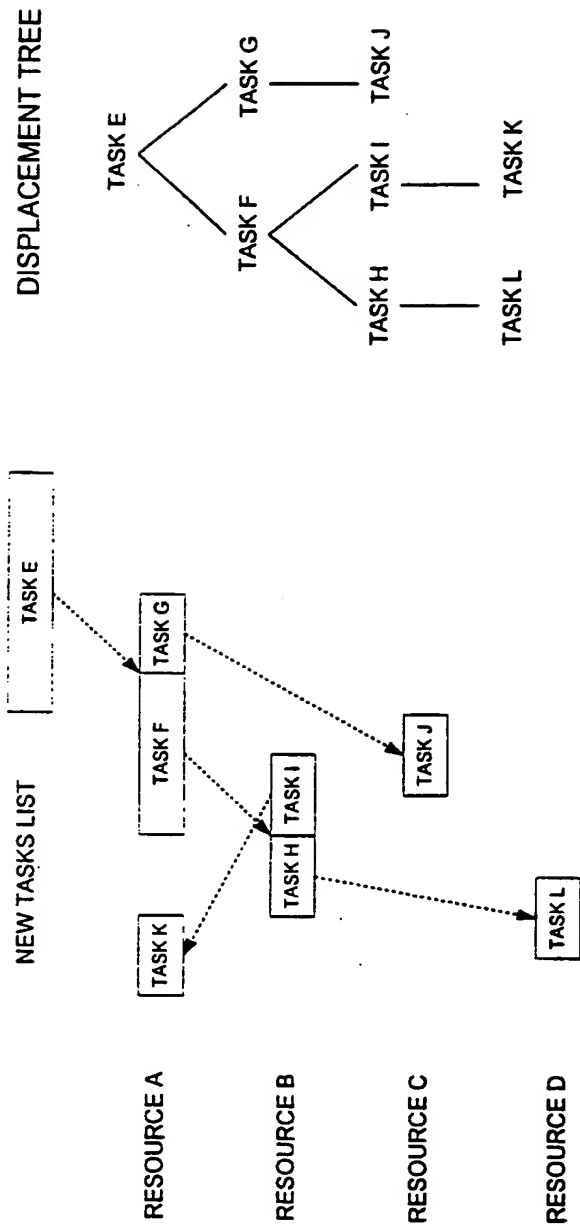


FIGURE 7B

FIGURE 7A

DISPLACED TASKS



FIGURE 7C

FIGURE 7D

8/8

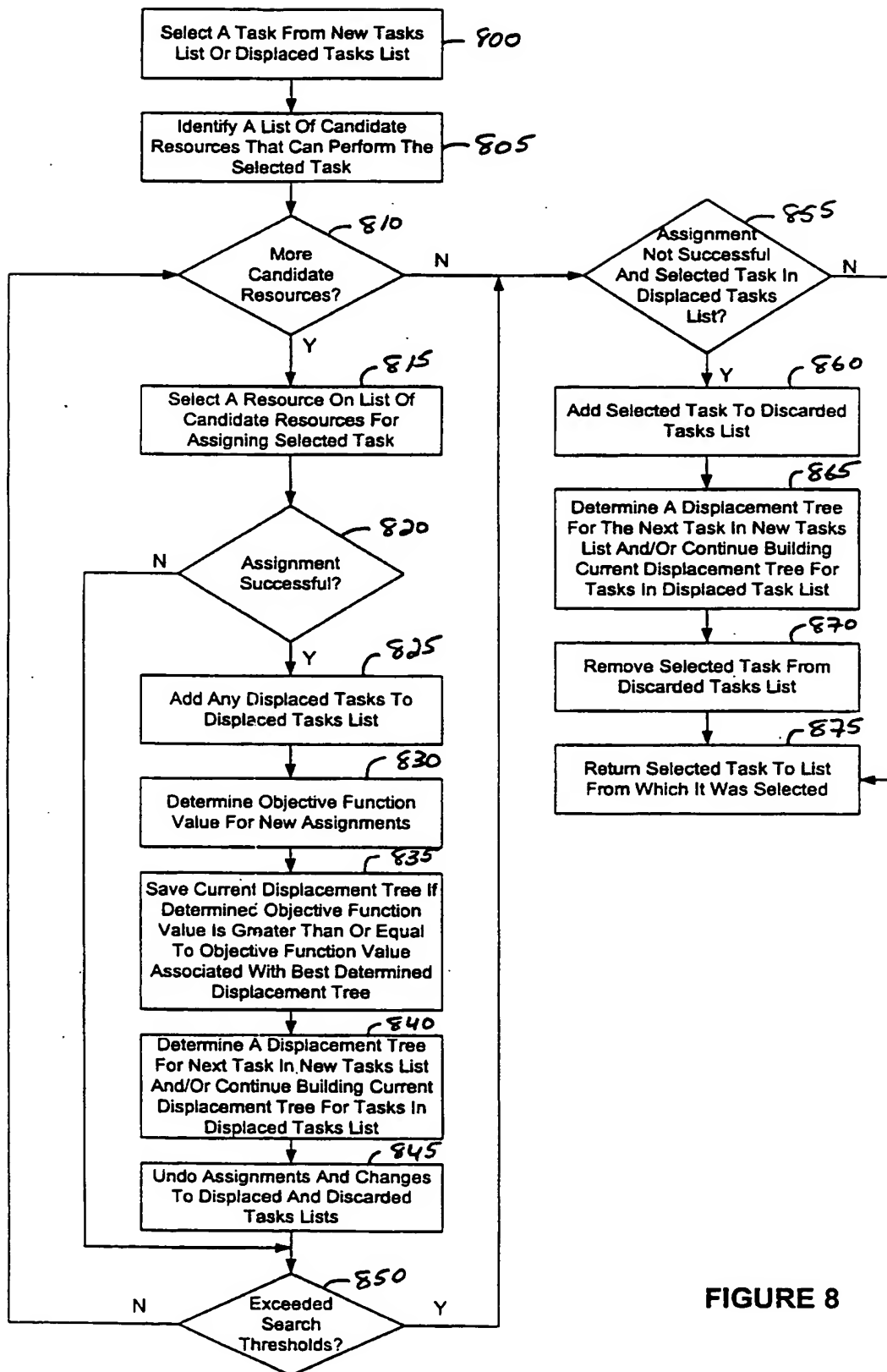


FIGURE 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/26301

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/60

US CL : 705/9

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/7,8,9

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Dialog, EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|-----------|---|-----------------------|
| A | Caseau, Y., "Using Constraint Propagation for Complex Scheduling Problems: Managing Size, Complex Resources and Travel", Lecture Notes in Computer Science, 1997, Vol. 1330, abstract only. | 1-7 |
| A | Caseau, Y., "Rule-Aided Constraint Resolution in Laure", Lecture Notes in Artificial Intelligence, 1991, Vol. 567, abstract only. | 1-7 |
| A | Shallcross, D., "neighbors of the Origin for 4 by 3 Matrices", Mathematics of Operations Research, 1992, Vol. 17, No. 3, abstract only. | 1-7 |



Further documents are listed in the continuation of Box C.



See patent family annex.

| | |
|---|--|
| * Special categories of cited documents: | "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention |
| "A" document defining the general state of the art which is not considered to be of particular relevance | "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone |
| "E" earlier document published on or after the international filing date | "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art |
| "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) | "&" document member of the same patent family |
| "O" document referring to an oral disclosure, use, exhibition or other means | |
| "P" document published prior to the international filing date but later than the priority date claimed | |

Date of the actual completion of the international search

27 NOVEMBER 2000

Date of mailing of the international search report

09 JAN 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

ERIC W. STAMPER

Telephone No. (703) 305-3900

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US00/26301

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|-----------|--|-----------------------|
| A | Caseau et al., "An Inventory Management Problem", Constraints, 1998 3(4), abstract only. | 1-7 |
| A | Caseau et al., "Solving Various Weighted Matching Problems with Constraints", Lecture Notes in Computer Science, 1997, Vol. 1330, abstract only. | 1-7 |
| A | Caseau et al., "Improving Branch and Bound for Jobshop Scheduling with Constraint Propagation", Lecture Notes in Computer Science, 1996, Vol. 1120, abstract only. | 1-7 |

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.